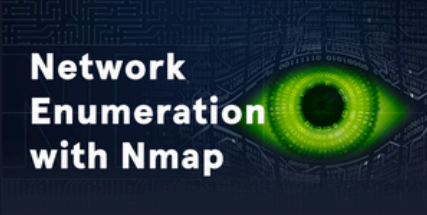


Nmap



Network Enumeration with Nmap

Nmap is one of the most used networking mapping and discovery tools because of its accurate results and efficiency. The tool is widely used by both offensive and defensive security practitioners. This module covers fundamentals that will be needed to use the Nmap tool for performing effective network enumeration.

OFFENSIVE EASY TIER I

COMPLETED 0%

Start

Enumération

Note

L'énumération est la partie la plus critique lors d'un pentest, il faut également comprendre comment fonctionne les services énumérés afin d'augmenter notre chance de réussir le pentest.

Introduction

Nmap est un outil Open-Source écrit en C, C++, Python et Lua.

Il est utilisé pour :

- Audit the security aspects of networks
- Simulate penetration tests
- Check firewall and IDS settings and configurations
- Types of possible connections
- Network mapping
- Response analysis
- Identify open ports
- Vulnerability assessment as well

```
nmap --help
```

<SNIP>

SCAN TECHNIQUES:

- sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
- sU: UDP Scan
- sN/sF/sX: TCP Null, FIN, and Xmas scans
- scanflags <flags>: Customize TCP scan flags
- sI <zombie host[:probeport]>: Idle scan
- sY/sZ: SCTP INIT/COOKIE-ECHO scans

```
-s0: IP protocol scan
-b <FTP relay host>: FTP bounce scan
```

Le plus utilisé est '**SS**' pour TCP, mais il ne faut pas oublié aussi d'utiliser '**SU**' pour UDP.

Host Discovery

Lors d'un pentest, la première chose à faire, est d'obtenir les hôtes présent dans le réseau, pour ce faire, il suffit de scanner l'adresse réseau :

```
sudo nmap 10.129.2.0/24 -sn -oA tnet | grep for | cut -d" " -f5
```

```
10.129.2.4
10.129.2.10
10.129.2.11
10.129.2.18
10.129.2.19
10.129.2.20
10.129.2.28
```

| Scanning Options | Description |
|------------------|-------------------------------------------------------------------------------------------------------------|
| 10.129.2.0/24 | Adresse réseau. |
| -sn | Pas de scanne de port. |
| -oA tnet | Stocke les résultats dans tous les formats commençant par le nom « tnet ». (moi je préfère tee hosts.txt) |

Scan IP liste

Ensuite, on peut scanner toutes les IP :

```
sudo nmap -sn -oA tnet -iL hosts.lst | grep for | cut -d" " -f5
```

```
10.129.2.18
10.129.2.19
10.129.2.20
```

| Scanning Options | Description |
|------------------|----------------------------------------------------------------------|
| -sn | Disables port scanning. |
| -oA tnet | Stores the results in all formats starting with the name 'tnet'. |
| -iL | Performs defined scans against targets in provided 'hosts.lst' list. |

Scan Multiple IPs

```
sudo nmap -sn -oA tnet 10.129.2.18 10.129.2.19 10.129.2.20 | grep for | cut -d" " -f5
```

```
10.129.2.18
10.129.2.19
10.129.2.20
```

Si ces adresses IP sont côte à côte, nous pouvons également définir la plage dans l'octet respectif :

```
sudo nmap -sn -oA tnet 10.129.2.18-20 | grep for | cut -d" " -f5

10.129.2.18
10.129.2.19
10.129.2.20
```

Scan Single IP

Avant d'analyser un hôte à la recherche de ports ouverts et de ses services, nous devons d'abord déterminer s'il est actif :

```
sudo nmap 10.129.2.18 -sn -oA host

Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-14 23:59 CEST
Nmap scan report for 10.129.2.18
Host is up (0.087s latency).
MAC Address: DE:AD:00:00:BE:EF
Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

Note

Par défaut, Nmap privilégie les requêtes ARP sur un réseau local. Pour observer un vrai ping ICMP, il est nécessaire d'ajouter l'option `-PE`, et l'on peut vérifier ce comportement avec `--packet-trace`.

```
sudo nmap 10.129.2.18 -sn -oA host -PE --packet-trace

Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-15 00:08 CEST
SENT (0.0074s) ARP who-has 10.129.2.18 tell 10.10.14.2
RCVD (0.0309s) ARP reply 10.129.2.18 is-at DE:AD:00:00:BE:EF
Nmap scan report for 10.129.2.18
Host is up (0.023s latency).
MAC Address: DE:AD:00:00:BE:EF
Nmap done: 1 IP address (1 host up) scanned in 0.05 seconds
```

| Scanning Options | Description |
|------------------|--------------------------------------------|
| 10.129.2.18 | Performs defined scans against the target. |
| -sn | Disables port scanning. |

| Scanning Options | Description |
|------------------|--------------------------------------------------------------------------|
| -oA host | Stores the results in all formats starting with the name 'host'. |
| -PE | Performs the ping scan by using 'ICMP Echo requests' against the target. |
| --packet-trace | Shows all packets sent and received |

une autre façon de déterminer si notre cible est 'vivante' est l'option --reason :

```
sudo nmap 10.129.2.18 -sn -oA host -PE --reason

Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-15 00:10 CEST
SENT (0.0074s) ARP who-has 10.129.2.18 tell 10.10.14.2
RCVD (0.0309s) ARP reply 10.129.2.18 is-at DE:AD:00:00:BE:EF
Nmap scan report for 10.129.2.18
Host is up, received arp-response (0.028s latency).
MAC Address: DE:AD:00:00:BE:EF
Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds
```

Pour désactiver les requêtes ARP et analyser notre cible avec les requêtes d'écho ICMP souhaitées, nous pouvons désactiver les pings ARP en définissant l'option « --disable-arp-ping » :

```
sudo nmap 10.129.2.18 -sn -oA host -PE --packet-trace --disable-arp-ping

Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-15 00:12 CEST
SENT (0.0107s) ICMP [10.10.14.2 > 10.129.2.18 Echo request (type=8/code=0) id=13607 seq=0] IP [ttl=255 id=23541 iplen=28 ]
RCVD (0.0152s) ICMP [10.129.2.18 > 10.10.14.2 Echo reply (type=0/code=0) id=13607 seq=0] IP [ttl=128 id=40622 iplen=28 ]
Nmap scan report for 10.129.2.18
Host is up (0.086s latency).
MAC Address: DE:AD:00:00:BE:EF
Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

Cela permet de passer en ICMP au lieu de ARP.

Note

Based on the last result, find out which operating system it belongs to. Submit the name of the operating system as result.

- **Linux/Unix** : TTL par défaut ≈ **64**
- **Windows** : TTL par défaut ≈ **128**
- **Cisco/Network devices** : TTL ≈ **255**

Answer : Windows

Host and Port Scanning

Scanner les 10 ports les plus utiliser :

```
sudo nmap 10.129.2.28 --top-ports=10
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-15 15:36 CEST
```

```
Nmap scan report for 10.129.2.28
```

```
Host is up (0.021s latency).
```

| PORT | STATE | SERVICE |
|----------|----------|---------------|
| 21/tcp | closed | ftp |
| 22/tcp | open | ssh |
| 23/tcp | closed | telnet |
| 25/tcp | open | smtp |
| 80/tcp | open | http |
| 110/tcp | open | pop3 |
| 139/tcp | filtered | netbios-ssn |
| 443/tcp | closed | https |
| 445/tcp | filtered | microsoft-ds |
| 3389/tcp | closed | ms-wbt-server |

MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)

```
Nmap done: 1 IP address (1 host up) scanned in 1.44 seconds
```

Nmap - Trace the Packets

```
sudo nmap 10.129.2.28 -p 21 --packet-trace -Pn -n --disable-arp-ping
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-15 15:39 CEST
```

```
SENT (0.0429s) TCP 10.10.14.2:63090 > 10.129.2.28:21 S ttl=56 id=57322 iplen=44  
seq=1699105818 win=1024 <mss 1460>
```

```
RCVD (0.0573s) TCP 10.129.2.28:21 > 10.10.14.2:63090 RA ttl=64 id=0 iplen=40  
seq=0 win=0
```

```
Nmap scan report for 10.11.1.28
```

```
Host is up (0.014s latency).
```

| PORT | STATE | SERVICE |
|--------|--------|---------|
| 21/tcp | closed | ftp |

MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)

```
Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
```

--packet-trace

Shows all packets sent and received.

Note

La ligne SENT montre que nous (10.10.14.2) avons envoyé un paquet TCP avec l'indicateur SYN (S) à notre cible (10.129.2.28). La ligne RCVD suivante montre que la cible répond avec un paquet TCP contenant les indicateurs RST et ACK (RA). Ces indicateurs servent à accuser réception du paquet TCP (ACK) et à mettre fin à la session TCP (RST).

Connect Scan

```
sudo nmap 10.129.2.28 -p 443 --packet-trace --disable-arp-ping -Pn -n --reason -sT
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-15 16:26 CET
CONN (0.0385s) TCP localhost > 10.129.2.28:443 => Operation now in progress
CONN (0.0396s) TCP localhost > 10.129.2.28:443 => Connected
Nmap scan report for 10.129.2.28
Host is up, received user-set (0.013s latency).
```

| PORT | STATE | SERVICE | REASON |
|---------|-------|---------|---------|
| 443/tcp | open | https | syn-ack |

```
Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
```

Note

le TCP connect Scan (-sT) utilise la négociation TCP pour déterminer si un port répond ou pas.

Pour ce faire, il envoie un paquet SYN sur le port cible, si la cible envoie en retour un paquet SYN-ACK, il répond. Sinon, ce sera un paquet RST.

Filtered Ports

Lorsqu'un port est indiqué comme **filtré**, cela peut avoir plusieurs causes. Dans la plupart des cas, cela signifie que des **règles de pare-feu** sont en place pour gérer certaines connexions. Les paquets peuvent être soit **supprimés (dropped)**, soit **rejetés (rejected)**.

Lorsque **un paquet est supprimé**, Nmap **ne reçoit aucune réponse** de la cible. Par défaut, le nombre de tentatives de renvoi (--max-retries) est fixé à **10**. Cela signifie que Nmap renverra la requête vers le port ciblé plusieurs fois pour déterminer si le paquet précédent a été perdu par erreur ou non.

Pour voir comment nos paquets sont traités :

```
sudo nmap 10.129.2.28 -p 139 --packet-trace -n --disable-arp-ping -Pn
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-15 15:45 CEST
SENT (0.0381s) TCP 10.10.14.2:60277 > 10.129.2.28:139 S ttl=47 id=14523 iplen=44
seq=4175236769 win=1024 <mss 1460>
SENT (1.0411s) TCP 10.10.14.2:60278 > 10.129.2.28:139 S ttl=45 id=7372 iplen=44
seq=4175171232 win=1024 <mss 1460>
Nmap scan report for 10.129.2.28
Host is up.
```

| PORT | STATE | SERVICE |
|---------|----------|-------------|
| 139/tcp | filtered | netbios-ssn |

MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)

```
Nmap done: 1 IP address (1 host up) scanned in 2.06 seconds
```

C'est la durée du Scan qui est différent, cela prend secondes au lieu de 0.05 secondes, il y a aussi des cas précis comme celui là :

```
sudo nmap 10.129.2.28 -p 445 --packet-trace -n --disable-arp-ping -Pn

Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-15 15:55 CEST
SENT (0.0388s) TCP 10.129.2.28:52472 > 10.129.2.28:445 S ttl=49 id=21763 iplen=44
seq=1418633433 win=1024 <mss 1460>
RCVD (0.0487s) ICMP [10.129.2.28 > 10.129.2.28 Port 445 unreachable
(type=3/code=3) ] IP [ttl=64 id=20998 iplen=72 ]
Nmap scan report for 10.129.2.28
Host is up (0.0099s latency).

PORT      STATE      SERVICE
445/tcp   filtered  microsoft-ds
MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 0.05 seconds
```

En réponse, nous recevons **un message ICMP de type 3 et de code d'erreur 3**, ce qui indique que **le port souhaité est injoignable**.

Discovering Open UDP Ports

Certains administrateurs oublient de filtrer les ports UDP.

UDP ne nécessite pas la négociation en 3 étapes, contrairement à TCP.

stateless protocol

Voici un exemple de scan UDP :

```
sudo nmap 10.129.2.28 -F -sU

Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-15 16:01 CEST
Nmap scan report for 10.129.2.28
Host is up (0.059s latency).
Not shown: 95 closed ports
PORT      STATE      SERVICE
68/udp    open|filtered dhcpc
137/udp   open          netbios-ns
138/udp   open|filtered netbios-dgm
631/udp   open|filtered ipp
5353/udp  open          zeroconf
MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 98.07 seconds
```

-F -> top 100 ports

```

sudo nmap 10.129.2.28 -sU -Pn -n --disable-arp-ping --packet-trace -p 137 --reason

Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-15 16:15 CEST
SENT (0.0367s) UDP 10.10.14.2:55478 > 10.129.2.28:137 ttl=57 id=9122 iplen=78
RCVD (0.0398s) UDP 10.129.2.28:137 > 10.10.14.2:55478 ttl=64 id=13222 iplen=257
Nmap scan report for 10.129.2.28
Host is up, received user-set (0.0031s latency).

PORT      STATE SERVICE      REASON
137/udp    open  netbios-ns    udp-response ttl 64
MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds

```

Si on a une réponse avec l'erreur code 3, c'est que le port est fermé :

```

Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-15 16:25 CEST
SENT (0.0445s) UDP 10.10.14.2:63825 > 10.129.2.28:100 ttl=57 id=29925 iplen=28
RCVD (0.1498s) ICMP [10.129.2.28 > 10.10.14.2 Port unreachable (type=3/code=3) ]
IP [ttl=64 id=11903 iplen=56 ]
Nmap scan report for 10.129.2.28
Host is up, received user-set (0.11s latency).

PORT      STATE SERVICE      REASON
100/udp    closed unknown port-unreach ttl 64
MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds

```

Version scan

L'option -sV permet d'afficher les versions des application présentent dans les ports ouverts :

```

sudo nmap 10.129.2.28 -Pn -n --disable-arp-ping --packet-trace -p 445 --reason -sV

Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-04 11:10 GMT
SENT (0.3426s) TCP 10.10.14.2:44641 > 10.129.2.28:445 S ttl=55 id=43401 iplen=44
seq=3589068008 win=1024 <mss 1460>
RCVD (0.3556s) TCP 10.129.2.28:445 > 10.10.14.2:44641 SA ttl=63 id=0 iplen=44
seq=2881527699 win=29200 <mss 1337>
NSOCK INFO [0.4980s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [0.4980s] nsock_connect_tcp(): TCP connection requested to
10.129.2.28:445 (IOD #1) EID 8
NSOCK INFO [0.5130s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS
for EID 8 [10.129.2.28:445]
Service scan sending probe NULL to 10.129.2.28:445 (tcp)
NSOCK INFO [0.5130s] nsock_read(): Read request from IOD #1 [10.129.2.28:445]
(timeout: 6000ms) EID 18
NSOCK INFO [6.5190s] nsock_trace_handler_callback(): Callback: READ TIMEOUT for
EID 18 [10.129.2.28:445]

```



```
Service scan sending probe SMBProgNeg to 10.129.2.28:445 (tcp)
NSOCK INFO [6.5190s] nsock_write(): Write request for 168 bytes to IOD #1 EID 27 [10.129.2.28:445]
NSOCK INFO [6.5190s] nsock_read(): Read request from IOD #1 [10.129.2.28:445] (timeout: 5000ms) EID 34
NSOCK INFO [6.5190s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 27 [10.129.2.28:445]
NSOCK INFO [6.5320s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 34 [10.129.2.28:445] (135 bytes)
Service scan match (Probe SMBProgNeg matched with SMBProgNeg line 13836):
10.129.2.28:445 is netbios-ssn. Version: |Samba smbd|3.X - 4.X|workgroup:
WORKGROUP|
NSOCK INFO [6.5320s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
Nmap scan report for 10.129.2.28
Host is up, received user-set (0.013s latency).
```

```
PORT      STATE SERVICE      REASON          VERSION
445/tcp open  netbios-ssn syn-ack ttl 63 Samba smbd 3.X - 4.X (workgroup:
WORKGROUP)
Service Info: Host: Ubuntu
```

```
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.55 seconds
```

Mise en pratique

```
nmap -sS -sV -Pn -sC -T5 10.129.102.194 -v
```

Find all TCP ports on your target. Submit the total number of found TCP ports as the answer :

7

Enumerate the hostname of your target and submit it as the answer. (case-sensitive) :

```
nix-nmap-default
```

Saving the Results

Different Formats

- Normal -> -oN
- Grepable -> oG
- XML -> oX

```
sudo nmap 10.129.2.28 -p- -oA target
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-16 12:14 CEST
Nmap scan report for 10.129.2.28
Host is up (0.0091s latency).
Not shown: 65525 closed ports
PORT      STATE SERVICE
```

```
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
```

```
MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)
```

```
Nmap done: 1 IP address (1 host up) scanned in 10.22 seconds
```

Note

Grâce à la sortie XML, nous pouvons facilement créer des rapports HTML faciles à lire, même pour les non-techniciens. Ceci est très utile pour la documentation, car les résultats sont présentés de manière détaillée et claire.

```
$ xsltproc target.xml -o target.html
```

Nmap Scan Report - Scanned at Tue Jun 16 12:14:53 2020

Scan Summary | 10.10.10.28

Scan Summary

Nmap 7.80 was initiated at Tue Jun 16 12:14:53 2020 with these arguments:
nmap -p- -oA target 10.10.10.28
Verbosity: 0; Debug level 0
Nmap done at Tue Jun 16 12:15:03 2020; 1 IP address (1 host up) scanned in 10.22 seconds

10.10.10.28

Address

- 10.10.10.28 (ipv4)
- DE:AD:00:00:BE:EF - Intel Corporate (mac)

Ports

| Port | State (toggle closed [0] filtered [0]) | | Service | Reason | Product | Version | Extra info |
|------|------------------------------------------|------|---------|---------|---------|---------|------------|
| 22 | tcp | open | ssh | syn-ack | | | |
| 25 | tcp | open | smtp | syn-ack | | | |
| 80 | tcp | open | http | syn-ack | | | |

Misc Metrics (click to expand)

| Metric | Value |
|--------------|--------------|
| Ping Results | arp-response |

Service Enumeration

-sV

```
sudo nmap 10.129.2.28 -p- -sV
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-15 19:44 CEST
```

```
[Space Bar]
```

```
Stats: 0:00:03 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
```

```
SYN Stealth Scan Timing: About 3.64% done; ETC: 19:45 (0:00:53 remaining)
```

--stats-every=5s permet de définir la durée d'affichage du statut :

```
sudo nmap 10.129.2.28 -p- -sV --stats-every=5s
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-15 19:46 CEST
```

```
Stats: 0:00:05 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
```

```
SYN Stealth Scan Timing: About 13.91% done; ETC: 19:49 (0:00:31 remaining)
```

```
Stats: 0:00:10 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 39.57% done; ETC: 19:48 (0:00:15 remaining)
```

Nmap analyse principalement les bannières des ports scannés et les imprime. S'il ne parvient pas à identifier les versions grâce à ces bannières, Nmap tente de les identifier grâce à un système de correspondance basé sur les signatures, ce qui rallonge considérablement la durée de l'analyse. L'un des inconvénients des résultats présentés par Nmap est que l'analyse automatique peut manquer certaines informations, car Nmap ne sait parfois pas les traiter.

Prenons un exemple :

```
sudo nmap 10.129.2.28 -p- -sV -Pn -n --disable-arp-ping --packet-trace

Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-16 20:10 CEST
<SNIP>
NSOCK INFO [0.4200s] nsock_trace_handler_callback(): Callback: READ SUCCESS for
EID 18 [10.129.2.28:25] (35 bytes): 220 inlane ESMTP Postfix (Ubuntu)..
Service scan match (Probe NULL matched with NULL line 3104): 10.129.2.28:25 is
smtp. Version: |Postfix smtpd||
NSOCK INFO [0.4200s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
Nmap scan report for 10.129.2.28
Host is up (0.076s latency).

PORT      STATE SERVICE VERSION
25/tcp    open  smtp      Postfix smtpd
MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)
Service Info: Host: inlane

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
```

La ligne :

```
NSOCK INFO [0.4200s] nsock_trace_handler_callback(): Callback: READ SUCCESS for
EID 18 [10.129.2.28:25] (35 bytes): 220 inlane ESMTP Postfix (Ubuntu)..
```

Note

Nous constatons ensuite que le serveur SMTP de notre cible nous a fourni plus d'informations que ce que Nmap nous a montré. Il s'agit ici de la distribution Linux Ubuntu. Cela se produit car, après une négociation à trois voies réussie, le serveur envoie souvent une bannière d'identification. Cela permet au client de savoir avec quel service il travaille. Au niveau réseau, cela se produit grâce à un indicateur PSH dans l'en-tête TCP.

Cependant, il peut arriver que certains services ne fournissent pas immédiatement ces informations. Il est également possible de supprimer ou de manipuler les bannières des services concernés. En nous connectant manuellement au serveur SMTP via nc, en récupérant la bannière et en interceptant le trafic réseau via tcpdump, nous pouvons voir ce que Nmap n'a pas montré :

```
sudo tcpdump -i eth0 host 10.10.14.2 and 10.129.2.28
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

```
1.| [SYN] | 18:28:07.128564 IP 10.10.14.2.59618 > 10.129.2.28.smtp: Flags [S],  
<SNIP> |  
2.| [SYN-ACK] | 18:28:07.255151 IP 10.129.2.28.smtp > 10.10.14.2.59618: Flags [S.],  
<SNIP> |  
3.| [ACK] | 18:28:07.255281 IP 10.10.14.2.59618 > 10.129.2.28.smtp: Flags [.],  
<SNIP> |
```

Note

Après cela, le serveur SMTP cible nous envoie un paquet TCP avec les indicateurs PSH et ACK, où PSH indique que le serveur cible nous envoie des données et avec ACK nous informe simultanément que toutes les données requises ont été envoyées.

```
4.| [PSH-ACK] | 18:28:07.319306 IP 10.129.2.28.smtp > 10.10.14.2.59618: Flags [P.],  
<SNIP> |
```

Note

Le dernier paquet TCP que nous avons envoyé confirme la réception des données avec un ACK.

```
5.| [ACK] | 18:28:07.319426 IP 10.10.14.2.59618 > 10.129.2.28.smtp: Flags [.],  
<SNIP> |
```

Nmap Scripting Engine

Il est possible avec NMAP de créer des scripts en Lua pour avoir des interactions avec certains services.

| Category | Description |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------|
| auth | Determination of authentication credentials. |
| broadcast | Scripts, which are used for host discovery by broadcasting and the discovered hosts, can be automatically added to the remaining scans. |
| brute | Executes scripts that try to log in to the respective service by brute-forcing with credentials. |
| default | Default scripts executed by using the <code>-sC</code> option. |
| discovery | Evaluation of accessible services. |
| dos | These scripts are used to check services for denial of service vulnerabilities and are used less as it harms the services. |

| Category | Description |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------|
| exploit | This category of scripts tries to exploit known vulnerabilities for the scanned port. |
| external | Scripts that use external services for further processing. |
| fuzzer | This uses scripts to identify vulnerabilities and unexpected packet handling by sending different fields, which can take much time. |
| intrusive | Intrusive scripts that could negatively affect the target system. |
| malware | Checks if some malware infects the target system. |
| safe | Defensive scripts that do not perform intrusive and destructive access. |
| version | Extension for service detection. |
| vuln | Identification of specific vulnerabilities. |

Default Scripts

```
sudo nmap <target> -sC
```

Specific Scripts Category

```
sudo nmap <target> --script <category>
```

Defined Script

```
sudo nmap <target> --script <script-name>,<script-name>,...
```

Nmap - Specifying Scripts

```
sudo nmap 10.129.2.28 -p 25 --script banner,smtp-commands
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-16 23:21 CEST
Nmap scan report for 10.129.2.28
Host is up (0.050s latency).
```

```
PORT      STATE SERVICE
25/tcp    open  smtp
|_banner: 220 inlane ESMTP Postfix (Ubuntu)
|_smtp-commands: inlane, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS,
ENHANCEDSTATUSCODES, 8BITMIME, DSN, SMTPUTF8,
MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)
```

Nmap - Aggressive Scan

```
sudo nmap 10.129.2.28 -p 80 -A
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-17 01:38 CEST
Nmap scan report for 10.129.2.28
Host is up (0.012s latency).
```

```

PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: WordPress 5.3.4
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: blog.inlanefreight.com
MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)
Warning: OSScan results may be unreliable because we could not find at least 1
open and 1 closed port
Aggressive OS guesses: Linux 2.6.32 (96%), Linux 3.2 - 4.9 (96%), Linux 2.6.32 -
3.10 (96%), Linux 3.4 - 3.10 (95%), Linux 3.1 (95%), Linux 3.2 (95%),
AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), Synology DiskStation
Manager 5.2-5644 (94%), Netgear RAIDiator 4.2.28 (94%),
Linux 2.6.32 - 2.6.35 (94%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1    11.91 ms  10.129.2.28

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.36 seconds

```

| -A | Performs service detection, OS detection, traceroute and uses defaults scripts to scan the target. |
|----|----------------------------------------------------------------------------------------------------|

Performance

Les performances d'analyse jouent un rôle important lorsqu'il s'agit d'analyser un réseau étendu ou de gérer une faible bande passante.

Timeouts

Lorsque Nmap envoie un paquet, il faut un certain temps (temps aller-retour - RTT) pour recevoir une réponse du port analysé.

Généralement, Nmap démarre avec un délai d'attente élevé (--min-RTT-timeout) de 100 ms. Prenons un exemple : l'analyse de l'ensemble du réseau avec 256 hôtes, dont les 100 premiers ports.

Default Scan

```

sudo nmap 10.129.2.0/24 -F

<SNIP>
Nmap done: 256 IP addresses (10 hosts up) scanned in 39.44 seconds

```

Optimized RTT

```
sudo nmap 10.129.2.0/24 -F --initial-rtt-timeout 50ms --max-rtt-timeout 100ms
```

<SNIP>

```
Nmap done: 256 IP addresses (8 hosts up) scanned in 12.29 seconds
```

| | |
|-----------------------------------------|-------------------------------------------------------|
| -F | Scans top 100 ports. |
| <code>--initial-rtt-timeout 50ms</code> | Sets the specified time value as initial RTT timeout. |
| <code>--max-rtt-timeout 100ms</code> | Sets the specified time value as maximum RTT timeout. |

Max Retries

```
--max-retries
```

Par défaut c'est à 10, mais on peut le descendre à 0, qui signifie que si le port n'envoie aucune réponse, nmap n'enverra pas plus de paquet.

Default Scan

```
sudo nmap 10.129.2.0/24 -F | grep "/tcp" | wc -l
```

```
23
```

Reduced Retries

```
sudo nmap 10.129.2.0/24 -F --max-retries 0 | grep "/tcp" | wc -l
```

```
21
```

| | |
|------------------------------|---------------------------------------------------------------------------|
| <code>--max-retries 0</code> | Sets the number of retries that will be performed during the scan. |
|------------------------------|---------------------------------------------------------------------------|

Rates

Lors d'un test d'intrusion en boîte blanche, nous pouvons être ajoutés à la liste blanche des systèmes de sécurité afin de détecter les vulnérabilités des systèmes du réseau, et pas seulement de tester les mesures de protection.

Connaître la bande passante du réseau permet de gérer le débit des paquets envoyés, ce qui accélère considérablement nos analyses avec Nmap.

Default Scan

```
sudo nmap 10.129.2.0/24 -F -oN tnet.default
```

<SNIP>

```
Nmap done: 256 IP addresses (10 hosts up) scanned in 29.83 seconds
```

Optimized Scan

```
sudo nmap 10.129.2.0/24 -F -oN tnet.minrate300 --min-rate 300
```

<SNIP>

Nmap done: 256 IP addresses (10 hosts up) scanned in 8.67 seconds

| | |
|--------------------------------------------------|------------------------------------------------------------------------|
| <code>-oN</code> <code>tnet.minrate300</code> | Saves the results in normal formats, starting the specified file name. |
| <code>--min-rate 300</code> | Sets the minimum number of packets to be sent per second. |

Default Scan - Found Open Ports

```
cat tnet.default | grep "/tcp" | wc -l
```

23

Optimized Scan - Found Open Ports

```
cat tnet.minrate300 | grep "/tcp" | wc -l
```

23

Timing

- `-T 0` / `-T paranoid`
- `-T 1` / `-T sneaky`
- `-T 2` / `-T polite`
- `-T 3` / `-T normal`
- `-T 4` / `-T aggressive`
- `-T 5` / `-T insane`

Insane Scan

```
sudo nmap 10.129.2.0/24 -F -oN tnet.T5 -T 5
```

<SNIP>

Nmap done: 256 IP addresses (10 hosts up) scanned in 18.07 seconds

| | |
|-------------------|---------------------------------------|
| <code>-T 5</code> | Specifies the insane timing template. |
|-------------------|---------------------------------------|

Firewall and IDS/IPS Evasion

NMAP offre plusieurs méthodes pour passer un IDS/IPS.

Ces méthodes incluent la fragmentation des paquets, l'utilisation de leurres et d'autres que nous aborderons dans cette section.

Determine Firewalls and Their Rules

Nous savons déjà que lorsqu'un port est affiché comme filtré, plusieurs raisons peuvent expliquer ce phénomène. Dans la plupart des cas, les pare-feu définissent des règles pour gérer des connexions spécifiques. Les paquets peuvent être soit ignorés, soit rejetés. Les paquets rejetés sont ignorés et l'hôte ne renvoie aucune réponse.

Il en va différemment des paquets rejetés, renvoyés avec un indicateur RST. Ces paquets contiennent différents types de codes d'erreur ICMP, voire aucun.

Les erreurs peuvent être :

- Net Unreachable
- Net Prohibited
- Host Unreachable
- Host Prohibited
- Port Unreachable
- Proto Unreachable

Note

La méthode d'analyse TCP ACK (-sA) de Nmap est beaucoup plus difficile à filtrer pour les pare-feu et les systèmes IDS/IPS que les analyses SYN (-sS) ou Connect (sT) classiques, car elles n'envoient qu'un paquet TCP avec l'indicateur ACK.

Avec -sS

```
sudo nmap 10.129.2.28 -p 21,22,25 -sS -Pn -n --disable-arp-ping --packet-trace

Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-21 14:56 CEST
SENT (0.0278s) TCP 10.10.14.2:57347 > 10.129.2.28:22 S ttl=53 id=22412 iplen=44
seq=4092255222 win=1024 <mss 1460>
SENT (0.0278s) TCP 10.10.14.2:57347 > 10.129.2.28:25 S ttl=50 id=62291 iplen=44
seq=4092255222 win=1024 <mss 1460>
SENT (0.0278s) TCP 10.10.14.2:57347 > 10.129.2.28:21 S ttl=58 id=38696 iplen=44
seq=4092255222 win=1024 <mss 1460>
RCVD (0.0329s) ICMP [10.129.2.28 > 10.10.14.2 Port 21 unreachable (type=3/code=3)
] IP [ttl=64 id=40884 iplen=72 ]
RCVD (0.0341s) TCP 10.129.2.28:22 > 10.10.14.2:57347 SA ttl=64 id=0 iplen=44
seq=1153454414 win=64240 <mss 1460>
RCVD (1.0386s) TCP 10.129.2.28:22 > 10.10.14.2:57347 SA ttl=64 id=0 iplen=44
seq=1153454414 win=64240 <mss 1460>
SENT (1.1366s) TCP 10.10.14.2:57348 > 10.129.2.28:25 S ttl=44 id=6796 iplen=44
seq=4092320759 win=1024 <mss 1460>
Nmap scan report for 10.129.2.28
Host is up (0.0053s latency).
```

```
PORT      STATE      SERVICE
21/tcp    filtered  ftp
22/tcp    open      ssh
25/tcp    filtered  smtp
MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
```

avec -sA

```
sudo nmap 10.129.2.28 -p 21,22,25 -sA -Pn -n --disable-arp-ping --packet-trace

Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-21 14:57 CEST
SENT (0.0422s) TCP 10.10.14.2:49343 > 10.129.2.28:21 A ttl=49 id=12381 iplen=40
seq=0 win=1024
SENT (0.0423s) TCP 10.10.14.2:49343 > 10.129.2.28:22 A ttl=41 id=5146 iplen=40
seq=0 win=1024
SENT (0.0423s) TCP 10.10.14.2:49343 > 10.129.2.28:25 A ttl=49 id=5800 iplen=40
seq=0 win=1024
RCVD (0.1252s) ICMP [10.129.2.28 > 10.10.14.2 Port 21 unreachable (type=3/code=3)
] IP [ttl=64 id=55628 iplen=68 ]
RCVD (0.1268s) TCP 10.129.2.28:22 > 10.10.14.2:49343 R ttl=64 id=0 iplen=40
seq=1660784500 win=0
SENT (1.3837s) TCP 10.10.14.2:49344 > 10.129.2.28:25 A ttl=59 id=21915 iplen=40
seq=0 win=1024
Nmap scan report for 10.129.2.28
Host is up (0.083s latency).

PORT      STATE      SERVICE
21/tcp    filtered  ftp
22/tcp    unfiltered ssh
25/tcp    filtered  smtp
MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds
```

| Scanning Options | Description |
|--------------------|---------------------------------------|
| 10.129.2.28 | Scans the specified target. |
| -p 21,22,25 | Scans only the specified ports. |
| -sS | Performs SYN scan on specified ports. |
| -sA | Performs ACK scan on specified ports. |
| -Pn | Disables ICMP Echo requests. |
| -n | Disables DNS resolution. |
| --disable-arp-ping | Disables ARP ping. |
| --packet-trace | Shows all packets sent and received. |

Detect IDS/IPS

Contrairement aux pare-feu et à leurs règles, la détection des systèmes IDS/IPS est beaucoup plus difficile, car il s'agit de systèmes de surveillance passive du trafic.

Il est recommandé d'utiliser plusieurs serveurs privés virtuels (VPS) avec différentes adresses IP afin de déterminer si de tels systèmes sont présents sur le réseau cible lors d'un test d'intrusion. Si l'administrateur détecte une telle attaque potentielle sur le réseau cible, la première étape consiste à bloquer l'adresse IP d'où provient l'attaque potentielle. Par conséquent, nous ne pourrons plus accéder au réseau avec cette adresse IP, et notre fournisseur d'accès à Internet (FAI) sera contacté et tout accès à Internet lui sera bloqué.

Une méthode pour déterminer la présence d'un tel système IPS sur le réseau cible consiste à analyser un seul hôte (VPS). Si cet hôte est bloqué et n'a pas accès au réseau cible, cela signifie que l'administrateur a pris des mesures de sécurité. Par conséquent, nous pouvons poursuivre notre test d'intrusion avec un autre VPS.

Decoys

Un autre exemple est celui où l'IPS doit nous bloquer. C'est pourquoi la méthode d'analyse des leurres (-D) est la plus adaptée. Avec cette méthode, Nmap génère diverses adresses IP aléatoires insérées dans l'en-tête IP afin de masquer l'origine du paquet envoyé. Cette méthode permet de générer aléatoirement (RND) un nombre spécifique (par exemple : 5) d'adresses IP séparées par deux points (:). Notre adresse IP réelle est ensuite placée aléatoirement entre les adresses IP générées. Dans l'exemple suivant, notre adresse IP réelle est donc placée en deuxième position. Autre point crucial : les leurres doivent être actifs. Sinon, le service cible risque d'être inaccessible en raison des mécanismes de sécurité SYN-flooding.

```
sudo nmap 10.129.2.28 -p 80 -sS -Pn -n --disable-arp-ping --packet-trace -D RND:5
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-21 16:14 CEST
SENT (0.0378s) TCP 102.52.161.59:59289 > 10.129.2.28:80 S ttl=42 id=29822
iplen=44 seq=3687542010 win=1024 <mss 1460>
SENT (0.0378s) TCP 10.10.14.2:59289 > 10.129.2.28:80 S ttl=59 id=29822 iplen=44
seq=3687542010 win=1024 <mss 1460>
SENT (0.0379s) TCP 210.120.38.29:59289 > 10.129.2.28:80 S ttl=37 id=29822
iplen=44 seq=3687542010 win=1024 <mss 1460>
SENT (0.0379s) TCP 191.6.64.171:59289 > 10.129.2.28:80 S ttl=38 id=29822 iplen=44
seq=3687542010 win=1024 <mss 1460>
SENT (0.0379s) TCP 184.178.194.209:59289 > 10.129.2.28:80 S ttl=39 id=29822
iplen=44 seq=3687542010 win=1024 <mss 1460>
SENT (0.0379s) TCP 43.21.121.33:59289 > 10.129.2.28:80 S ttl=55 id=29822 iplen=44
seq=3687542010 win=1024 <mss 1460>
RCVD (0.1370s) TCP 10.129.2.28:80 > 10.10.14.2:59289 SA ttl=64 id=0 iplen=44
seq=4056111701 win=64240 <mss 1460>
Nmap scan report for 10.129.2.28
Host is up (0.099s latency).

PORT      STATE SERVICE
80/tcp    open  http
```

MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds

**-D
RND:5**

Generates five random IP addresses that indicates the source IP the connection comes from.

Un autre scénario serait que seuls certains sous-réseaux n'aient pas accès aux services spécifiques du serveur. Nous pouvons donc spécifier manuellement l'adresse IP source (-S) pour vérifier si nous obtenons de meilleurs résultats. Les leurres peuvent être utilisés pour les analyses SYN, ACK, ICMP et de détection de système d'exploitation. Prenons un exemple et déterminons quel système d'exploitation est le plus susceptible d'être concerné.

```
sudo nmap 10.129.2.28 -n -Pn -p445 -O
```

Starting Nmap 7.80 (<https://nmap.org>) at 2020-06-22 01:23 CEST

Nmap scan report for 10.129.2.28

Host is up (0.032s latency).

| PORT | STATE | SERVICE |
|------|-------|---------|
|------|-------|---------|

| | | |
|---------|----------|--------------|
| 445/tcp | filtered | microsoft-ds |
|---------|----------|--------------|

MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)

Too many fingerprints match this host to give specific OS details

Network Distance: 1 hop

OS detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 3.14 seconds

avec -S

```
sudo nmap 10.129.2.28 -n -Pn -p 445 -O -S 10.129.2.200 -e tun0
```

Starting Nmap 7.80 (<https://nmap.org>) at 2020-06-22 01:16 CEST

Nmap scan report for 10.129.2.28

Host is up (0.010s latency).

| PORT | STATE | SERVICE |
|------|-------|---------|
|------|-------|---------|

| | | |
|---------|------|--------------|
| 445/tcp | open | microsoft-ds |
|---------|------|--------------|

MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Aggressive OS guesses: Linux 2.6.32 (96%), Linux 3.2 - 4.9 (96%), Linux 2.6.32 - 3.10 (96%), Linux 3.4 - 3.10 (95%), Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), Synology DiskStation Manager 5.2-5644 (94%), Linux 2.6.32 - 2.6.35 (94%), Linux 2.6.32 - 3.5 (94%)

No exact OS matches for host (test conditions non-ideal).

Network Distance: 1 hop

OS detection performed. Please report any incorrect results at

<https://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 4.11 seconds

| | |
|--------------|---------------------------------------------------------------|
| -S | Scans the target by using different source IP address. |
| 10.129.2.200 | Specifies the source IP address. |
| -e tun0 | Sends all requests through the specified interface. |

DNS Proxying

Nmap permet néanmoins de spécifier nous-mêmes les serveurs DNS (`--dns-server <ns>, <ns>`). Cette méthode peut s'avérer essentielle si nous sommes dans une zone démilitarisée (DMZ). Les serveurs DNS de l'entreprise sont généralement plus fiables que ceux d'Internet. Par exemple, nous pouvons les utiliser pour interagir avec les hôtes du réseau interne. À titre d'exemple, nous pouvons utiliser le port TCP 53 comme port source (`--source-port`) pour nos analyses. Si l'administrateur utilise le pare-feu pour contrôler ce port et ne filtre pas correctement les IDS/IPS, nos paquets TCP seront approuvés et transmis.

Filtered Port

```
sudo nmap 10.129.2.28 -p50000 -sS -Pn -n --disable-arp-ping --packet-trace
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-21 22:50 CEST
```

```
SENT (0.0417s) TCP 10.10.14.2:33436 > 10.129.2.28:50000 S ttl=41 id=21939  
iplen=44 seq=736533153 win=1024 <mss 1460>
```

```
SENT (1.0481s) TCP 10.10.14.2:33437 > 10.129.2.28:50000 S ttl=46 id=6446 iplen=44  
seq=736598688 win=1024 <mss 1460>
```

```
Nmap scan report for 10.129.2.28
```

```
Host is up.
```

```
PORT      STATE      SERVICE  
50000/tcp filtered ibm-db2
```

```
Nmap done: 1 IP address (1 host up) scanned in 2.06 seconds
```

DNS port specified

```
sudo nmap 10.129.2.28 -p50000 -sS -Pn -n --disable-arp-ping --packet-trace --  
source-port 53
```

```
SENT (0.0482s) TCP 10.10.14.2:53 > 10.129.2.28:50000 S ttl=58 id=27470 iplen=44  
seq=4003923435 win=1024 <mss 1460>
```

```
RCVD (0.0608s) TCP 10.129.2.28:50000 > 10.10.14.2:53 SA ttl=64 id=0 iplen=44  
seq=540635485 win=64240 <mss 1460>
```

```
Nmap scan report for 10.129.2.28
```

```
Host is up (0.013s latency).
```

```
PORT      STATE SERVICE  
50000/tcp open  ibm-db2
```

```
MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)
```

```
Nmap done: 1 IP address (1 host up) scanned in 0.08 seconds
```

```
--source-port 53
```

Performs the scans from specified source port.

Maintenant que nous avons découvert que le pare-feu accepte le port TCP 53, il est fort probable que les filtres IDS/IPS soient configurés avec une puissance beaucoup plus faible que les autres.

Connexion au port filtré :

```
ncat -nv --source-port 53 10.129.2.28 50000
```

```
Ncat: Version 7.80 ( https://nmap.org/ncat )
```

```
Ncat: Connected to 10.129.2.28:50000.
```

```
220 ProFTPd
```

Firewall and IDS/IPS Evasion Easy Labs

Énoncé

Une entreprise nous a mandatés pour tester ses défenses informatiques, notamment ses systèmes IDS et IPS. Notre client souhaite renforcer sa sécurité informatique et apportera donc des améliorations spécifiques à ses systèmes IDS/IPS après chaque test réussi. Nous ne connaissons cependant pas les directives qui seront suivies pour ces modifications. Notre objectif est d'obtenir des informations spécifiques à partir de situations données.

Nous disposons uniquement d'une machine protégée par des systèmes IDS/IPS et pouvons être testés. À des fins d'apprentissage et pour comprendre le comportement des IDS/IPS, nous avons accès à une page web d'état à l'adresse suivante :

```
http://<target>/status.php
```

Our client wants to know if we can identify which operating system their provided machine is running on. Submit the OS name as the answer.

```
nmap -sSCV -Pn IP
```

```
PORT      STATE SERVICE
22/tcp    open  ssh
| ssh-hostkey:
|   2048 71c189907ffd4f60e054f385e6356c2b (RSA)
|   256 e18e531842af2adec0121e2e54064f70 (ECDSA)
|_  256 1accacd4945cd61d71e739de14273c3c (ED25519)
80/tcp    open  http
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
|_ http-title: Apache2 Ubuntu Default Page: It works
10001/tcp open  scp-config
```

```
NSE: Script Post-scanning.
Initiating NSE at 22:05
Completed NSE at 22:05, 0.00s elapsed
Initiating NSE at 22:05
Completed NSE at 22:05, 0.00s elapsed
```

```
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 76.55 seconds
Raw packets sent: 4 (152B) | Rcvd: 1 (28B)
```

Firewall and IDS/IPS Evasion - Medium Lab

Énoncé

Après avoir effectué le premier test et soumis nos résultats à notre client, les administrateurs ont apporté quelques modifications et améliorations à l'IDS/IPS et au pare-feu. Lors de la réunion, nous avons constaté qu'ils n'étaient pas satisfaits de leurs configurations précédentes, et ils ont constaté que le trafic réseau pouvait être filtré plus strictement.

After the configurations are transferred to the system, our client wants to know if it is possible to find out our target's DNS server version. Submit the DNS server version of the target as the answer.

```
nmap -sSCV -Pn -p 53 10.129.2.48 -D RND:5 -v
```

```
PORT      STATE SERVICE VERSION
53/tcp    open  domain (unknown banner: HTB{GoTtgUnyze9Psw4vGjcuMpHRp})
| dns-nsid:
|_  bind.version: HTB{GoTtgUnyze9Psw4vGjcuMpHRp}
| fingerprint-strings:
|_  DNSVersionBindReqTCP:
|      version
|      bind
|_  HTB{GoTtgUnyze9Psw4vGjcuMpHRp}
```

Firewall and IDS/IPS Evasion - Hard Lab

Enoncé

Grâce à notre deuxième test, notre client a pu acquérir de nouvelles connaissances et a envoyé l'un de ses administrateurs suivre une formation sur les systèmes IDS/IPS. Comme il nous l'avait indiqué, la formation durerait une semaine. L'administrateur a pris toutes les précautions nécessaires et souhaite que nous testions à nouveau cette formation, car certains services doivent être modifiés, ainsi que la communication du logiciel fourni.

```
sudo nmap -sS -p- -Pn -n -p- --source-port 53 10.129.2.47
```

Note

On se fait passer par le DNS, afin de tromper le pare-feux ou IPS/IDS

- -n désactive la résolution DNS
- -Pn désactive le ping
- -sS SYN scan
- --source-port 53 on se fait passer pour le port du DNS
- -p- Scanner tout les ports

le port 50 000 était ouvert

nc 10.129.2.47 50000 -p 53

HTB{kjnsdf2n982n1827eh76238s98di1w6}